homepage.html

* /employee/getall
* /employee/searchbyname
* /employee/insert

/employee/searchbyname

homepage.html -> search by name or deptno

輸入員工姓名或部門編號進行模糊搜尋

把名字相似的找出來

使用fuse.js

getall -> queryall.ejs

update:

delete:  
1. 在button使用data-empno，並用onclick=”deleteData()”

* 在 HTML 中，data-\* 是一組自定義數據屬性（custom data attributes），用來在 HTML 元素上儲存自定義數據。這些屬性不會影響元素的顯示或行為，但可以由 JavaScript 存取，常用於傳遞資料給前端腳本。
* 具體來說，data-empno 是你為某個 HTML 元素（例如按鈕）定義的一個自定義屬性，用來儲存與該元素相關的數據（在你的情境中，是員工編號 empno）。
* 例如，在你的 queryall.ejs 中，你提到為刪除按鈕添加 <button data-empno="<%= emp.empno %>" onclick="deleteData(this)">，這裡的 data-empno 會將員工的 empno 值綁定到按鈕上。

為什麼需要 data-empno？

* 在實現刪除功能中，每筆員工資料在表格中有一個對應的刪除按鈕。你需要知道用戶點擊哪個按鈕時要刪除哪個員工的資料。
* 因為每個按鈕對應不同的員工（由 empno 唯一識別），你需要在按鈕上儲存 empno，讓 JavaScript 的 deleteData 函數能夠獲取這個值，並發送給後端以刪除對應的資料。
* data-empno 提供了一個簡單的方式，將 empno 綁定到按鈕上，方便 JavaScript 透過 dataset 屬性或 getAttribute() 取出。

2. 後端從 json 陣列中移除對應的員工資料（使用 filter()）

let result = json.filter(emps => emps.empno !== empno);

3. json to csv

[How to Convert JSON to CSV File in JavaScript?](https://www.tutorialspoint.com/how-to-convert-json-to-csv-file-in-javascript)

參考這個寫法用map()把json轉回去csv

const headers = Object.keys(result[0]);

這行程式碼假設 result 是一個陣列，且陣列的第一個元素（result[0]）是一個物件（例如 { empno: "1001", ename: "潘麗珍", ... }）。

Object.keys() 會返回物件的所有鍵（key），例如 ["empno", "ename", "hiredate", "salary", "deptno", "title"]。

這是用來自動獲取 CSV 的標題行（headers），但前提是 result 不為空，且所有物件的鍵一致（結構相同）

const headers = Object.keys(result[0]);

* 這行程式碼假設 result 是一個陣列，且陣列的第一個元素（result[0]）是一個物件（例如 { empno: "1001", ename: "潘麗珍", ... }）。
* Object.keys() 會返回物件的所有鍵（key），例如 ["empno", "ename", "hiredate", "salary", "deptno", "title"]。
* 這是用來自動獲取 CSV 的標題行（headers），但前提是 result 不為空，且所有物件的鍵一致（結構相同）。

const rows = result.map(obj => headers.map(header => obj[header]).join(','));

* 這行使用 map() 遍歷 result 陣列中的每個物件（obj）。
* 對於每個 obj，使用 headers.map(header => obj[header]) 獲取對應的鍵值（例如 obj["empno"]、obj["ename"] 等），生成一個陣列（如 ["1001", "潘麗珍", "2010-11-10", "56000", "100", "senior engineer"]）。
* 然後用 join(',') 將這個陣列轉為逗號分隔的字串（例如 "1001,潘麗珍,2010-11-10,56000,100,senior engineer"）。
* 最終，rows 是一個陣列，包含所有轉換後的 CSV 行（例如 ["1001,潘麗珍,2010-11-10,56000,100,senior engineer", "1002,林麗如,2008-03-22,34000,100,engineer", ...]）。

最後我用的轉換寫法是  
// json to csv

    const header = "empno,ename,hiredate,salary,deptno,title";

    let csvContent = header + '\n';

    if (result.length > 0) {

        const rows = result.map(emps =>

            `${emps.empno},${emps.ename},${emps.hiredate},${emps.salary},${emps.deptno},${emps.title}`

        );

        csvContent += rows.join('\n'); // 加入資料行

    }

    console.log(csvContent);

    fs.writeFile(file\_path, csvContent, (err) => {

        if(err) throw err;

        console.log('Saved!');

    });

4. 使用 fs.writeFile() 將更新後的 json 轉換為 CSV 格式，寫回 employee.csv

5. 確保刪除後，前端能更新顯示（例如刷新頁面或移除表格行）

homepage.html -> insert.ejs

1. emp.js的insert部分
2. 用get接到insert.ejs
3. 用post執行5件事:
4. 建立function檢查所有欄位是否填寫
5. 建立function檢查 empno 是否是 4 位數字
6. 建立function檢查 empno 是否已存在
7. 產生 CSV 格式的新員工資料
8. 加資料到 CSV
9. insert.ejs

用<% %>來接後端傳來的message跟status

<% if (message && message !== '') { %>

        <p class="message <%= status %>" id="msg"><%= message %></p>

<% } %>